EER and EER for Pharmacy

- For ER
  - 1-1, 1-m, m-n, and m-1 relationships
  - Attribute definitions on entities - more than described.
  - Primary MD as relationship
  - Missing RX number from Prescription Entity
- For EER
  - Use of extended ER constructs for inheritance
  - Who can prescribe has changed
  - Relationship between Prescriber and Patient
  - Limit “Who” can prescribe to one of the options (use disjoint)
Notice **Movement** of Common Attributes to Person and DEA number to Prescriber

“Person” is a parent of both Patient and Physician and Physician also a Sub-entity of Drug Prescriber

---

**Problem 3.19**

1. The database represents each AIRPORT, keeping its unique AirportCode, the AIRPORT Name, and the City and State in which the AIRPORT is located.
2. Each airline FLIGHT has a unique number, the Airline for the FLIGHT, and the Weekdays on which the FLIGHT is scheduled (for example, every day of the week except Sunday can be coded as X7).
3. A FLIGHT is composed of one or more FLIGHT LEGs (for example, flight number CO1223 from New York to Los Angeles may have two FLIGHT LEGs: leg 1 from New York to Houston and leg 2 from Houston to Los Angeles). Each FLIGHT LEG has a DEPARTURE AIRPORT and Scheduled Departure Time, and an ARRIVAL AIRPORT and Scheduled Arrival Time.
4. A LEG INSTANCE is an instance of a FLIGHT LEG on a specific Date (for example, CO1223 leg 1 on July 30, 1989). The actual Departure and Arrival AIRPORTs and Times are recorded for each flight leg after the flight leg has been concluded. The Number of available seats and the AIRPLANE used in the LEG INSTANCE are also kept.
5. The customer RESERVATIONs on each LEG INSTANCE include the Customer Name, Phone, and Seat Number(s) for each reservation.
6. Information on AIRPLANEs and AIRPLANE TYPES are also kept. For each AIRPLANE TYPE (for example, DC-10), the Type Name, manufacturing Company, and Maximum Number of Seats are kept. The AIRPORTs in which planes of this type CAN LAND are kept in the database. For each AIRPLANE, the Airplanefield, Total number of seats, and TYPE are kept.
Problem 3.23

• a. Entity types: BANK, ACCOUNT, CUSTOMER, LOAN

• b. Weak entity type: BANK-BRANCH. Partial key: BranchNo. Identifying relationship: BRANCHES.

• c. The partial key BranchNo in BANK-BRANCH specifies that the same BranchNo value may occur under different BANKs. The identifying relationship BRANCHES specifies that BranchNo values are uniquely assigned for those BANK-BRANCH entities that are related to the same BANK entity. Hence, the combination of BANK Code and BranchNo together constitute a full identifier for a BANK-BRANCH.

Problem 3.23

d. Relationship Types: BRANCHES, ACCTS, LOANS, A-C, L-C. The (min, max) constraints are shown below.
Problem 3.23

e. The requirements may be stated as follows: Each BANK has a unique Code, as well as a Name and Address. Each BANK is related to one or more BANK-BRANCHes, and the BranchNo is unique among each set of BANK-BRANCHes that are related to the same BANK. Each BANK-BRANCH has an Address. Each BANK-BRANCH has zero or more LOANS and zero or more ACCTS. Each ACCOUNT has an AcctNo (unique), Balance, and Type and is related to exactly one BANK-BRANCH and to at least one CUSTOMER. Each LOAN has a LoanNo (unique), Amount, and Type and is related to exactly one BANK-BRANCH and to at least one CUSTOMER. Each CUSTOMER has an SSN (unique), Name, Phone, and Address, and is related to zero or more ACCOUNTs and to zero or more LOANs.

Problem 7.23a 3rd/6.18a 4th & 5th

How many copies of the book titled The Lost Tribe are owned by the library branch whose name is ‘Sharpstown’?

\[
\pi_{\text{No\_Of\_Copies}} (\sigma_{\text{BranchName}=\text{Sharpstown}} (\text{LIBRARY-BRANCH})) \times \pi_{\text{BookId}} (\text{BOOKCOPIES} \times (\sigma_{\text{Title}=\text{The Lost Tribe}} (\text{BOOK})))
\]
Homework 2 Solution – 40 pt total

- For Problem 2.1 - 7.23/6.18 – a, b, c, and d
  - No points for 7.23a/6.18a – solution posted on web
  - 6 pts each for b and c
  - 8 pts for d – see point breakdown on slide 2
  - Note – show using P(rojection), S(elect), *(join)
  - Or the equivalent Greek letters

- For Problem 2.2 - 7.25/6.20 – a, b, and c
  - 5pts a, 3 pts b, 4 pts c

- For Problem 2.3 - 7.27/6.22 – a, b, e, and f only
  - 2 pts per table for each a, b, e, and f

Problem 7.23 3rd/6.18 4th & 5th – a to d

a. P No_Of_Copies (( s BranchName='Sharpstown' (LIBRARY-BRANCH)) * 
  (BOOKCOPIES * ( s Title='The Lost Tribe' (BOOK))))

\[ \pi_{\text{No Of Copies}}(\sigma_{\text{BranchName='Sharpstown'}}(\text{LIBRARY-BRANCH})) \times \]
\[ \text{BranchID} \]
\[ (\text{BOOKCOPIES} \times (\sigma_{\text{Title='The Lost Tribe'}}(\text{BOOK}))) \]

b. P BranchID, No_Of_Copies ( ( s Title='The Lost Tribe' (BOOK)) 
  * BOOKCOPIES * LIBRARY_BRANCH )

\[ \pi_{\text{BranchName, No Of Copies}}((\sigma_{\text{Title='The Lost Tribe'}}(\text{BOOK})) \times \]
\[ \text{BookId} \]
\[ \text{BranchId} \]

2 pts each BookID and BranchID joins
1pt select, 1pt project
Problem 7.23 3rd/6.18 4th & 5th – a to d

CSE 4701

Problem 7.25 3rd/6.20 4th & 5th – a to c

CSE 4701
Problem 7.27 3rd/6.22 4th & 5th – a to g

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>10</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
<td>6</td>
<td>25</td>
<td>c</td>
<td>3</td>
</tr>
</tbody>
</table>

b.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
<td>10</td>
<td>b</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
</tbody>
</table>

c.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>b</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>a</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>b</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>c</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>b</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

d.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a</td>
<td>5</td>
<td>10</td>
<td>b</td>
<td>5</td>
</tr>
</tbody>
</table>
Homework 3 Solution

First two Problems – Write SQL

- Homework Problem 3.1:
  - 3rd, 4th, and 5th Editions: Problem 8.11 from the textbook and Problem 6.18 from the 6th edition but do this in SQL and NOT relational calculus.

- Homework Problem 3.2:

Problem 8.11/6.18 in 6th edition

a. SELECT NoOfCopies
   FROM ( (BOOK NATURAL JOIN BOOK_COPIES )
           NATURAL JOIN LIBRARY_BRANCH )
   WHERE Title='The Lost Tribe' AND BranchName='Sharpstown'

b. SELECT BranchName, NoOfCopies
   FROM ( (BOOK NATURAL JOIN BOOK_COPIES )
           NATURAL JOIN LIBRARY_BRANCH )
   WHERE Title='The Lost Tribe'

c. SELECT Name
   FROM BORROWER B
   WHERE NOT EXIST ( SELECT *
                      FROM BOOK_LOANS L
                      WHERE B.CardNo = L.CardNo )

d. SELECT B.Title, R.Name, R.Address
   FROM BOOK B, BORROWER R, BOOK_LOANS BL, LIBRARY_BRANCH LB
   WHERE LB.BranchName='Sharpstown' AND LB.BranchId=BL.BranchId AND
   BL.DueDate='today' AND BL.CardNo=R.CardNo AND BL.BookId=B.BookId
Problem 8.11/6.18 in 6th edition

c. SELECT L.BranchName, COUNT(*)
FROM BOOK_COPIES B, LIBRARY_BRANCH L
WHERE B.BranchId = L.BranchId
GROUP BY L.BranchName

f. SELECT B.CardNo, B.Name, B.Address, COUNT(*)
FROM BORROWER B, BOOK_LOANS L
WHERE B.CardNo = L.CardNo
GROUP BY B.CardNo
HAVING COUNT(*) > 5

g. SELECT Title, NoOfCopies
FROM ((BOOK_AUTHORS NATURAL JOIN BOOK)
       NATURAL JOIN BOOK_COPIES)
       NATURAL JOIN LIBRARY_BRANCH)
WHERE Author_Name = 'Stephen King' and BranchName = 'Central'

Problem 8.13/4.10 in 6th edition

(a) Retrieve the names of employees in department 5 who
work more than 10 hours per week on the 'ProductX' project.
SELECT LNAME, FNAME
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE DNO=5 AND SSN=ESSN AND
PNO=PNUMBER AND PNAME='ProductX' AND HOURS>10

SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE DNO=5 AND SSN IN
  ( SELECT ESSN
    FROM WORKS_ON
    WHERE HOURS>10 AND PNO IN
      ( SELECT PNUMBER
        FROM PROJECT
        WHERE PNAME='ProductX' ) )

  LNAME   FNAME
  Smith    John
  English  Joyce
Problem 8.13/4.10 in 6th edition

(b) List the names of employees who have a dependent with the same first name as themselves.

```
SELECT LNAME, FNAME
FROM EMPLOYEE, DEPENDENT
WHERE SSN=ESSN AND FNAME=DEPENDENT_NAME
```

Another possible SQL query uses nesting as follows:

```
SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE EXISTS
    ( SELECT * 
      FROM DEPENDENT 
      WHERE FNAME=DEPENDENT_NAME 
        AND SSN=ESSN )
```

Result (empty):

![Employee names](LNAME FNAME)

Smith John
Narayan Ramesh
English Joyce

Problem 8.13/4.10 in 6th edition

(c) Find the names of employees that are directly supervised by 'Franklin Wong'.

```
SELECT E.LNAME, E.FNAME
FROM EMPLOYEE E, EMPLOYEE S
WHERE S.FNAME='Franklin' AND 
    S.LNAME='Wong' AND E.SUPERSSN=S.SSN
```

Another possible SQL query uses nesting as follows:

```
SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE SUPERSSN IN
    ( SELECT SSN 
      FROM EMPLOYEE 
      WHERE FNAME='Franklin' AND LNAME='Wong' )
```
(d) For each project, list the project name and the total hours per week (by all employees) spent on that project.

```
SELECT PNAME, SUM(HOURS)
FROM PROJECT, WORKS_ON
WHERE PNUMBER=PNO
GROUP BY PNAME
```

Result:

<table>
<thead>
<tr>
<th>PNAME</th>
<th>SUM(HOURS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductX</td>
<td>52.5</td>
</tr>
<tr>
<td>ProductY</td>
<td>37.5</td>
</tr>
<tr>
<td>ProductZ</td>
<td>50.0</td>
</tr>
<tr>
<td>Computerization</td>
<td>55.0</td>
</tr>
<tr>
<td>Reorganization</td>
<td>25.0</td>
</tr>
<tr>
<td>Newbenefits</td>
<td>55.0</td>
</tr>
</tbody>
</table>

(e) Retrieve the names of employees who work on every project.

```
SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE NOT EXISTS
  ( SELECT PNUMBER
    FROM PROJECT
    WHERE NOT EXISTS
      ( SELECT *
        FROM WORKS_ON
        WHERE PNUMBER=PNO AND ESSN=SSN ) )
```

Result (empty):
(f) Retrieve the names of employees who do not work on any project.

```
SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE NOT EXISTS
    ( SELECT *
        FROM WORKS_ON
        WHERE ESSN=SSN )
```

Result (empty):

(g) For each department, retrieve the department name, and the average salary of employees working in that department.

```
SELECT DNAME, AVG(SALARY)
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO
GROUP BY DNAME
```

Result:

<table>
<thead>
<tr>
<th>DNAME</th>
<th>AVG(SALARY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>33250</td>
</tr>
<tr>
<td>Administration</td>
<td>31000</td>
</tr>
<tr>
<td>Headquarters</td>
<td>55000</td>
</tr>
</tbody>
</table>
Problem 8.13/4.10 in 6th edition

(h) Retrieve the average salary of all female employees.
SELECT AVG (SALARY)
FROM EMPLOYEE
WHERE SEX='F'

Result:
AVG(SALARY)
31000

(i) Find the names and addresses of employees who work
on at least one project located in Houston but
whose department has no location in Houston.
SELECT LNAME, FNAME, ADDRESS
FROM EMPLOYEE
WHERE EXISTS
( SELECT *
  FROM WORKS_ON, PROJECT
  WHERE SSN=ESSN AND PNO=PNUMBER
  AND PLOCATION='Houston'
)
AND NOT EXISTS
( SELECT *
  FROM DEPT_LOCATIONS
  WHERE DNO=DNUMBER
  AND DLOCATION='Houston'
)

Result:
LNAME   FNAME ADDRESS
Wallace Jennifer 291 Berry, Bellaire, TX
(j) List the last names of department managers who have no dependents.
SELECT LNAME, FNAME
FROM EMPLOYEE
WHERE EXISTS
    ( SELECT *
        FROM DEPARTMENT
        WHERE SSN=MGRSSN )
AND NOT EXISTS
    ( SELECT *
        FROM DEPENDENT
        WHERE SSN=ESSN )

Result:
LNAME  FNAME
Borg    James
Homework Problem 3.3: For the Chinook Database Schema
a. Find the list of all Customers and Employees that have the same last name and print out the Last Name (only once), Address, City, and State for each.

```
SELECT employee.LastName, employee.Address, employee.City, employee.State, customer.Address, customer.city, customer.state
FROM chinook.employee, chinook.customer
WHERE employee.LastName = customer.LastName;
```

b. Find and print the Customer Names (First and Last) and Company Name of all Customers that have purchased a Rock Album. (where the Name is ‘Rock’ in the Genre table).

```
SELECT DISTINCT customer.FirstName, customer.LastName, customer.Company
FROM chinook.customer, chinook.invoice, chinook.invoiceline, chinook.track, chinook.genre
WHERE customer.CustomerId = invoice.CustomerId
AND invoice.InvoiceId = invoiceline.InvoiceId AND invoiceline.TrackId = track.TrackId
AND track.GenreId = genre.GenreId AND genre.name = 'rock';
```

c. Find all of print album name and tracks of all of the albums by the composer James Hetfield, grouped by Album.

```
SELECT album.title, track.name
FROM chinook.album, chinook.track
WHERE album.AlbumId = track.AlbumId AND track.Composer LIKE '%Hetfield%';
```

d. For each customer that lives in Canada (CA- Country attribute of Customer), find all invoices and for each result, and print Customer Last Name, number of invoices for customer, and the total amount paid for all invoices.

```
SELECT customer.LastName, COUNT(invoice.InvoiceId), SUM(invoice.total)
FROM chinook.customer, chinook.invoice
WHERE customer.Country = 'Canada' AND customer.CustomerId = invoice.CustomerId
GROUP BY customer.LastName;
```
Homework Problem 3.4: For the Northwind Database Schema

a. Find and print the company names and company addresses of all Suppliers that supply the category name Seafood.

```
SELECT DISTINCT suppliers.CompanyName, suppliers.Address
FROM northwind.suppliers, northwind.categories, northwind.products
WHERE suppliers.SupplierID = products.SupplierID
AND categories.CategoryID = products.CategoryID
AND categories.CategoryName = 'Seafood';
```

b. Count and print the number of suppliers for each of the eight different categories of food which by name are: Beverages, Condiments, Confections, Dairy Products, Grains/Cereals, Meat/Poultry, Produce, Seafood.

```
SELECT categories.CategoryName, COUNT(suppliers.SupplierID)
FROM northwind.categories, northwind.products, northwind.suppliers
WHERE suppliers.SupplierID = products.SupplierID
AND products.categoryID = categories.CategoryID
GROUP BY categories.CategoryName;
```

c. For each country (ShipCountry in Orders), total the Freight charges. The countries are: France, Germany, Brazil, Belgium, Switzerland, Venezuela, Austria, Mexico, USA, Sweden, Finland, Italy, Spain, UK, Ireland, Portugal, Canada, Denmark, Poland, Norway, Argentina

```
SELECT orders.ShipCountry, SUM(orders.Freight)
FROM northwind.orders
GROUP BY orders.ShipCountry;
```
Homework 4 Solution

AIRPLANE(Reg#, Model, Number)
   pk   fk   fk

PLANE_TYPE(Model, Capacity, Weight)
   pk

FLIES(Model, SSN)

WORKS_ON(Model, SSN)

HANGER(Number, Capacity, Location)
   pk

OWNS(Reg#, OwnerId, Pdate)

OWNER(OwnerId)
   pk

MAINTAIN(Workcode, Date, Reg#, SSN)

CORPORATION(Name, Address, Phone, OwnerId)
   pk   fk

PERSON(SSN, Name, Address, Phone, OwnerId, PFlag, Lic_Num, Restr,
   pk   fk   EFlag, Salary, Shift)

SERVICE(Workcode, Date, Reg#, Hours)
   pk